Name: Ali Bilal
Roll No: P20-0077
Section: BSC-6C

LAB TASK 3

In [44]:
```
# Data manipulation imports
import numpy as np
import pandas as pd

# Visualization imports
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline

# Modeling imports
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, ConfusionMatrixDisplay, confusion_matrix, classification_report
```

Generating Synthetic Data for a Binary Classification Problem

In [45]:
```
np.random.seed(0)

df = pd.DataFrame({'X1': np.random.randint(1, 10, size=50),
                   'X2': np.random.randint(3, 10, size=50),
                   'Y': np.random.choice(['Bad', 'Good'], size=50)})
df
```

In [51]:
```
plt.figure(figsize=(10,6))
plt.scatter(df['X1'], df['X2'], c=df['Y'].apply(lambda x: 'red' if x == 'Bad' else 'green'), marker='o')
plt.xlabel('X1: Acid Durability',fontsize=14)
plt.ylabel('X2: Strength',fontsize=14)
plt.show()
```

```
df
```

In [46]:
```
df = df.sample(frac=1).reset_index(drop=True)
```

In [47]:
```
df.head()
```

In [ ]:
```
new_data = pd.DataFrame({'X1': [6,1,3], 'X2': [5,2,3]})

new_label = knn.predict(new_data)
print("Predicted Label for New Data:", new_label)
```

In [ ]:
```
# print(confusion_matrix(y_test,y_pred))
```

In [64]:
```
import pandas as pd
from sklearn.model_selection import train_test_split
```

In [65]:
```
path ="fruit_data_with_colors _1_.csv"
data = pd.read_csv(path)
data.head(5)
```

Out[65]:

	fruit_label	fruit_name	fruit_subtype	mass	width	height	color_score
0	1	apple	granny_smith	192.0	8.4	7.3	0.55
1	1	apple	granny_smith	180.0	8.0	6.8	0.59
2	1	apple	granny_smith	176.0	7.4	7.2	0.60
3	2	mandarin	mandarin	86.0	6.2	4.7	0.80
4	2	mandarin	mandarin	84.0	6.0	4.6	0.79

In [ ]:
```
y_pred= knn.predict(X_test)
y_pred
```

Evaluating Model Performance with Accuracy Score

The accuracy score is calculated by comparing the true labels of the test set (y_test) with the predicted
accuracy score is a commonly used metric for evaluating classification models, as it measures the prop

In [ ]:
```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

In [52]:
```
# Split the data into training and testing sets
X = df[['X1', 'X2']]
y = df['Y']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
print(X_test)
print(y_test)
```

In [88]:
```
n_neighbors_list = [1, 3, 5, 7, 9]

# Initialize an empty list to store the accuracy scores
accuracy_scores = []

# Iterate over the values and fit the KNN model for each value
for n_neighbors in n_neighbors_list:
    knn = KNeighborsClassifier(n_neighbors=n_neighbors)
    knn.fit(X_train, y_train)
    y_pred = knn.predict(X_test)
    acc = accuracy_score(y_test, y_pred)
    accuracy_scores.append(acc)

# Plot the accuracy scores
plt.plot(n_neighbors_list, accuracy_scores, '-o')
plt.xlabel('Value of K')
plt.ylabel('Accuracy score')
plt.title('Accuracy score for different values of K')
plt.show()
```